PHYS 232 – Assignment #6

Due Friday, Apr. 5 @ 11:00

For a function $y(x)$ that is linear in the parameters a_k:

$$y(x) = \sum_k a_k f_k(x),$$

where $f_k(x)$ are any functions of x that do not involve the parameters a_k, we showed that minimizing χ^2 led to a relatively simple expression that could be used to determine the best-fit parameters from a set of data $(x_i,\ y_i \pm \sigma_i)$ for i from 1 to N:

$$\underline{a} = \underline{\underline{\varepsilon}}\ \underline{\beta}.$$

Here, $\underline{a}$ is a column matrix of the unknown parameters, $\underline{\beta}$ is a column matrix with elements given by:

$$\beta_k = \sum_{i=1}^{N} \frac{y_i f_k(x_i)}{\sigma_i^2},$$

and $\underline{\underline{\varepsilon}}$ is a symmetric square matrix determined from $\underline{\underline{\varepsilon}} = \underline{\underline{\alpha}}^{-1}$. The symmetric square matrix $\underline{\underline{\alpha}}$ has elements given by:

$$\alpha_{\ell k} = \sum_{i=1}^{N} \left[\frac{1}{\sigma_i^2} f_\ell(x_i) f_k(x_i) \right].$$

The diagonal elements of the *error* matrix $\underline{\underline{\varepsilon}}$ give the square of the uncertainties in the best-fit parameters, such that:

$$\sigma_{a_k} = \sqrt{\varepsilon_{kk}}.$$

In this assignment, you will be given sets of data and you are to determine the best-fit parameters and their uncertainties using the method outlined above. I recommend that you use Python to evaluate the necessary sums and to complete the necessary matrix algebra. You can find example Python code that shows how to construct and manipulate matrices here: `https://people.ok.ubc.ca/jbobowsk/Python.html`. Here is a link that will open the matrices tutorial in UBC's Open Jupyter.

1. The following table shows calibration data from a type–J thermocouple.

T (°C)	V (mV)
300 ± 2	16.327
350 ± 2	19.090
400 ± 2	21.848
450 ± 3	24.610
500 ± 3	27.393
550 ± 3	30.216
600 ± 3	33.102
650 ± 4	36.071
700 ± 4	39.132
750 ± 4	42.281
800 ± 4	45.494

Over this wide temperature range the T versus V data are not linearly related but can be fit to:

$$T = a_1 + a_2 V + a_3 V^2.$$

Use the methods describe on page one to determine the best-fit values of $a_1 \pm \sigma_{a_1}$, $a_2 \pm \sigma_{a_2}$, and $a_3 \pm \sigma_{a_3}$ from the given set of data. Plot the T versus V data (with error bars) and show the best-fit curve.

Next, use Python to fit the data to a second degree polynomial. Make sure that your fit is a *weighted* fit. You can find an example weighted polynomial fit here: `https://people.ok.ubc.ca/jbobowsk/Python.html`. Again, here is a direct link that will open the polynomial tutorial in UBC's Open Jupyter.

Compare the parameters and uncertainties determined by Python to those that you calculated. You should find that the a_k values are very similar. However, I found that the error estimates determined from the diagonal elements of $\underline{\underline{\varepsilon}}$ are significantly larger than the uncertainties reported by Python. Bonus marks to anyone that can explain how Python is determining the uncertainties and why they are smaller than those that we calculated. The number of bonus marks awarded will be determined by the completeness and clarity of your explanation!

2. One model that applies to a wide variety of physical situations is:

$$y(t) = a_1 + (a_2 - a_1)e^{-t/\tau}. \tag{1}$$

(i) If charging a capacitor from some initial voltage V_0 to a final voltage V_b using a battery and a simple series RC circuit, then Eq. 1 describes the voltage across the capacitor as a function of time:

$$V(t) = V_\mathrm{b} + (V_0 - V_\mathrm{b})\, e^{-t/\tau},$$

where $\tau = RC$ is the time constant of the circuit.

(ii) If you throw an object off of a tall building with initial speed v_0 and the drag force can be modelled as $F_\mathrm{d} = -bv$, then the speed of the object as a function of time is given by:

$$v(t) = v_\mathrm{T} + (v_0 - v_\mathrm{T})\, e^{-t/\tau},$$

where v_T is the terminal velocity of the object and the time constant τ is determined from the object's mass and the drag coefficient b.

(iii) Newton's law of cooling approximately describes the time evolution of the temperature of an object, with initial temperature T_0, placed in an environment with temperature T_rm:

$$T(t) = T_\mathrm{rm} + (T_0 - T_\mathrm{rm})\, e^{-t/\tau}. \tag{2}$$

Here, the time constant τ will, in part, be determined by the heat capacity of the object. This last example is interesting because it can be used to estimate the time of death of a recently-deceased person. For an average body τ and T_0 are reasonably well known (T_0 is the average body temperature of a living person) and T_rm and $T(t)$ are easily measured. An initial estimate of the time t that has elapsed since death can then be determined.

Your task for problem #2 is given on the next page. You will need to complete a "linear-in-parameters" fit to determine a set of best-fit parameters and their uncertainties. To complete this problem, you can either use the method outlined on the first page (i.e. start by constructing the $\underline{\underline{\alpha}}$ and $\underline{\beta}$ matrices), or you can simply do the fit in Python. You don't have to do both methods (as you were asked to do in the first problem).

(a) Put Eq. 2 into the form $T(t) = T_{\rm rm} f_1(t) + T_0 f_2(t)$. What are $f_1(t)$ and $f_2(t)$? Notice that $T(t)$ is linear in the parameters $T_{\rm rm}$ and T_0, but not in τ. For the remainder of this problem we will assume that the time constant $\tau = 50$ hr is a known constant.

(b) The Riddler, an archenemy of Batman, has ruthlessly killed an innocent person. Next to the body Batman finds the following note:

Batman, I have kidnapped your man-servant Alfred Pennyworth. If you cannot solve the riddle below, he will be the next to die. You have until Apr. 3, 2024 at 11:00 to get me your answer.

After killing this innocent victim that I left for you, I recorded the body temperature as a function of time since death and tabulated the data:

time (hours)	Temp. (°C)
6	36.5±0.4
10	34.8±0.4
14	33.9±0.4
18	32.7±0.3
22	30.8±0.3
26	30.4±0.3
30	29.5±0.2
34	28.3±0.2
38	27.6±0.2
42	26.5±0.2
46	26.2±0.2
50	25.4±0.2
54	24.8±0.2
58	24.1±0.2
62	23.6±0.2

You must determine the initial temperature of the victim's body at the time of death and the temperature of the room in which the victim was killed. Use the known body-cooling time constant $\tau = 50$ hrs and include uncertainties with your solution. Plot the data with error bars and the best-fit line. Do you have what it takes to save Pennyworth?

Practice Problem – won't be graded

For a function $y(x)$ that is linear in parameters a_k:

$$y(x) = \sum_k a_k f_k(x),$$

where $f_k(x)$ are any functions of x that do not involve the parameters a_k, we showed that minimizing χ^2 led to $\underline{a} = \underline{\underline{\varepsilon}}\,\underline{\beta}$, a relatively simple expression that can be used to determine the best-fit parameters from a set of data $(x_i,\ y_i \pm \sigma_i)$ for i from 1 to N. The ℓ^{th} element of $\underline{a}$ can be expressed as:

$$a_\ell = \sum_{k=1} \varepsilon_{\ell k}\beta_k, \tag{3}$$

where:

$$\beta_k = \sum_{i=1}^{N} \frac{y_i f_k(x_i)}{\sigma_i^2}, \tag{4}$$

and $\underline{\underline{\varepsilon}}$ is a symmetric square matrix determined from $\underline{\underline{\varepsilon}} = \underline{\underline{\alpha}}^{-1}$. The symmetric square matrix $\underline{\underline{\alpha}}$ has elements given by:

$$\alpha_{\ell k} = \sum_{i=1}^{N} \left[\frac{1}{\sigma_i^2} f_\ell(x_i) f_k(x_i)\right]. \tag{5}$$

For the case of $y(x) = a_1 + a_2 x$, use Eqs. (3) – (5) to show that $a_1 \pm \sigma_1$ and $a_2 \pm \sigma_2$ are given by the expressions that we previously derived for weighted linear fits. Namely:

$$a_1 = \frac{1}{\Delta}\left[\sum \frac{x_i^2}{\sigma_i^2} \sum \frac{y_i}{\sigma_i^2} - \sum \frac{x_i}{\sigma_i^2} \sum \frac{x_i y_i}{\sigma_i^2}\right], \tag{6a}$$

$$a_2 = \frac{1}{\Delta}\left[\sum \frac{1}{\sigma_i^2} \sum \frac{x_i y_i}{\sigma_i^2} - \sum \frac{x_i}{\sigma_i^2} \sum \frac{y_i}{\sigma_i^2}\right], \tag{6b}$$

$$\sigma_1^2 = \frac{1}{\Delta} \sum \frac{x_i^2}{\sigma_i^2}, \tag{6c}$$

$$\sigma_2^2 = \frac{1}{\Delta} \sum \frac{1}{\sigma_i^2}, \tag{6d}$$

where all of the sums are from $i = 1$ to N and:

$$\Delta \equiv \sum \frac{1}{\sigma_i^2} \sum \frac{x_i^2}{\sigma_i^2} - \left(\sum \frac{x_i}{\sigma_i^2}\right)^2. \tag{7}$$

This proves, of course, that linear fits are just a special case of the more general fits to functions of the form $y(x) = \sum_k a_k f_k(x)$.